Power Converter

6.331 Lab 2
Piotr Mitros

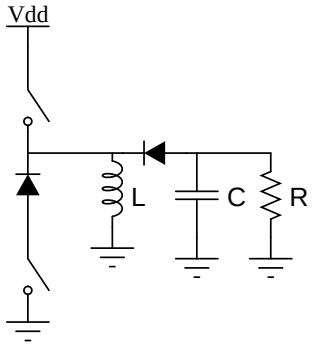
February 22, 2002

1 Introduction

We came up with a somewhat nifty topology, ran out of time in the implementation, mostly because in addition to the interesting topology, we were attempting to make the output of the power converter power several of the devices, and had some startup issues (the circuit had a number of other fairly ambitious aspects, including feedforward control of several parameters based on input voltage level, a very high switching frequency, etc.). As a result, we implemented a boring, brute-force approach to meet the specs. We'll present the desired (interesting) topology at the beginning of the paper, and mention what we did instead at the end.

2 The VAL Power Converter

Our basic power converter topology is:



Here, we have three cycles. During D_1, the inductor is connected to V_{DD} and charging, D_2 when it is connected to ground, storing charge, and D_3 when it is discharging on the load.

This system gives us the characteristic equations:

$$\tilde{i}Ls = D_1 \cdot V_{in} - D_3 \cdot V_{out}$$

$$\tilde{v}_{out}Cs = -\frac{V_{out}}{R} + D_3 I$$

Linearizing,

$$\tilde{i}Ls = (D_1 + d_1)V_{IN} - (D_3 + d_3) \cdot (V_{OUT} + \tilde{v}_{out})$$
$$\tilde{v}_{out}Cs = -\frac{V_{OUT}}{R} - \frac{\tilde{v}_{out}}{R} + D_3 I + d_3 I + D_3\tilde{i}$$

Simplifying,

$$\tilde{v}_{out}Cs = -\frac{\tilde{v}_{out}}{R} + d_3 I + D_3\tilde{i}$$
$$\tilde{i}Ls = d_1 V_{IN} - D_3\tilde{v}_{out} - d_3 V_{OUT}$$

And combining,

$$\tilde{v}_{out}Cs = -\frac{\tilde{v}_{out}}{R} + d_3 I + \frac{D_3}{Ls}\left(d_1 V_{IN} - D_3\tilde{v}_{out} - d_3 V_{OUT}\right)$$

Grouping $\tilde{v}_{out}$,

$$\tilde{v}_{out}(Cs + \frac{1}{R} + \frac{D_3^2}{Ls}) = d_3 I + \frac{D_3}{Ls}\left(d_1 V_{IN} - d_3 V_{OUT}\right)$$

$$\tilde{v}_{out}(LCs^2 + \frac{Ls}{R} + D_3^2) = (ILs - D_3 V_{OUT})d_3 + (D_3 V_{IN})d_1$$

$$\tilde{v}_{out} = \frac{ILs - D_3 V_{OUT}}{LCs^2 + \frac{Ls}{R} + D_3^2}d_3 + \frac{D_3 V_{IN}}{LCs^2 + \frac{Ls}{R} + D_3^2}d_1$$

There are two obvious ways to control this:

- We can now open-loop control D_3 and closed-loop control D_1. This eliminates the loop gain dependence of V_{IN}, but we still keep a right half-plane zero.
- We can closed-loop control D_3. We get no zero, but we maintain load dependence.

However, we can do significantly better if we can control both. We define a control variable ζ and let $d_1 = \psi\zeta$ and $d_3 = \zeta$, for $\psi > \frac{V_{OUT}}{V_{IN,min}}$. This gives:

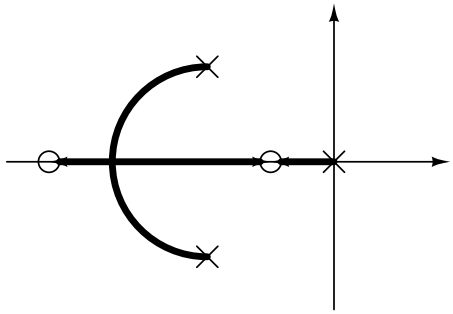
$$\tilde{v}_{out} = \frac{ILs - D_3 V_{OUT}}{LCs^2 + \frac{Ls}{R} + D_3^2}\zeta + \frac{D_3 V_{IN}}{LCs^2 + \frac{Ls}{R} + D_3^2}\psi\zeta = \frac{Ls - D_3 V_{OUT} + \psi D_3 V_{IN}}{LCs^2 + \frac{Ls}{R} + D_3^2}\zeta$$

With $\psi D_3 V_{IN} - D_3 V_{OUT} > 0$. This moves the zero from the right half plane into the left half plane. Physically, what happens is when we adjust D_3, the output drops before it begins to rise. We exploit this by adjusting D_3 in reverse. We use it to give a little bit of extra boost at first, but by the time the output would start falling off from D_3, D_1 kicks in to compensate. The zero is very far out, for small swings of D_3. For large swings of D_3, the power converter is somewhat inefficient.

Practically, this means we drive overall duty cycle down by a fair margin if we want to increase output. This short duty cycle results in a much greater strain on the devices, although we can, again, reduce this in three ways:

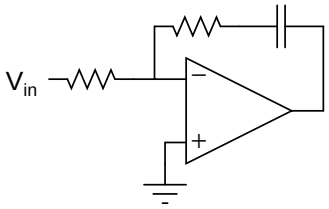
- Since this system is already very responsive, we can lower switching frequency
- We can open loop control to set the time of D_2 based on the current V_{IN}. This sets the baseline for D_1 and D_3 at a given V_{IN} to a reasonable value, while keeping the $D_1 : D_3$ ratio fixed.
- We can distribute the load among several devices

From there, we can compensate it with a pole at the origin and a second zero in the left half plane. This gives us root-locus of:



To a first order, we can extend the gain arbitrarily high, until we run into the higher-order poles around the switching frequency, op amp poles, etc.

Our circuit for this is:



We were planning to run a second feedback loop around this to keep the center value reasonable, so that we would only have a small period of time idleing.

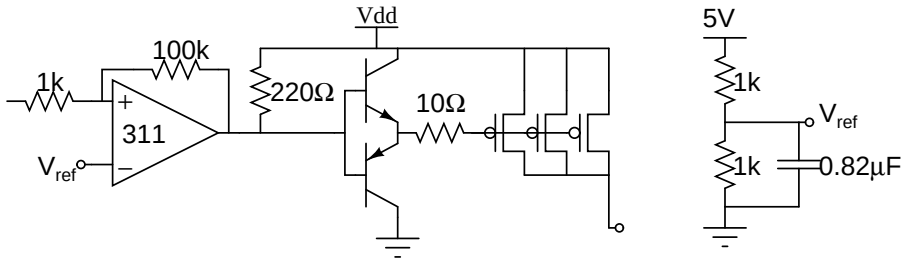
3 Our Converter

We used a standard boost topology for a -10V output. In order to increase switching frequency, we distributed the switching load among several switches.

3.1 High Voltage Stages

Unless otherwise noted, V_{DD} refers to the input voltage in the high voltage stages.

3.2 Switches



We found that the IRFD9110s had a somewhat higher R_{ON} than rated. Our power dissipation was actually dominated by static dissipation. Worse, as the switches warmed up, R_{ON} would incease, and we would have severe thermal runway problems. To combat this problem, we stacked three devices in parallel. This reduced static dissipation to be completely manageable.

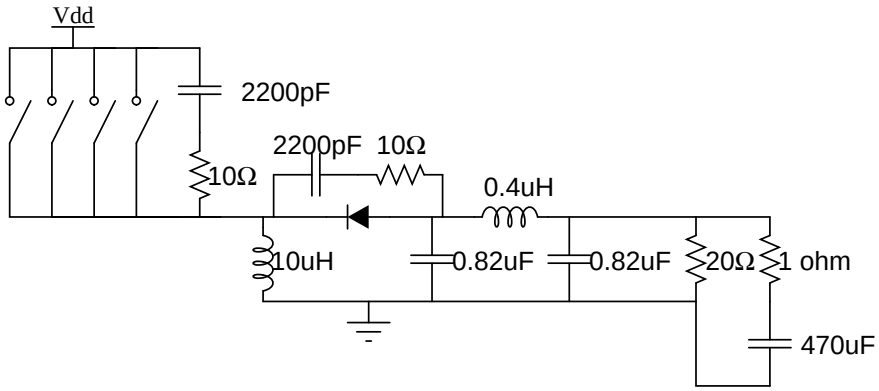
Each switch is driven by a 3904/3906 push-pull pair. That, in turn, is driven by an LM311 comparator. The comparator compares the TTL output of the previous stage to a 2.5V reference voltage. The same reference is used for all switches.

On all 311s, all the extra pins on the V_+ side are connected to the input voltage, and all on the V_- side are tied to ground. According to the spec sheet, this gives slightly higher output currents.

For power dissipation calculations, we initially used $P = I_l^2 R T_1$ for static. We approximated dynamic as $\tau_{switch} \cdot (V_{IN} - V_{OUT}) \cdot I_l \cdot f/2$. This assumes current drawn stays fixed, while output voltage is a triangle wave. In addition, we have some additional power dissipation due to switching oscillations, while some of the switching power is eaten by the damping leg. We did not calculate those. With the specified R_{ON} from the spec sheets, and the power split among the switches, we found that we could conservatively switch at about 2MHz. In practice, due to R_{ON} issues, we scaled this down to 1MHz.

3.3 Output Stage

Our output stage is as follows:



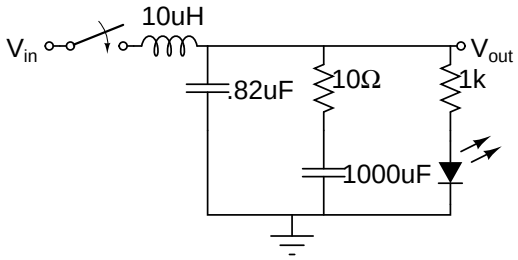
During the design stage, we chose the 1.6pf of output capacitance is chosen to keep switching ripple under $200mV$ ($C = I_{out} T_3 / V_{ripple}$). We found that in practice, although this managed ripple, we still had a large spike due to some switching transient, so we did

not meet our ripple spec. To improve this, we added damping legs on the diode and on the transistors. In addition, we added a small inductor to filter out high frequency spikes (LC time constant of about). This gives about $175mV$ of ripple including switching transient, and $60mV$ before.

We chose $L = V_{OUT}T_3/I_l$, so that the inductor current would not have time to fully discharge. This gave a value of about $10mH$, but we increased it a bit in order to be conservative.

We use a single damping leg for all of the switches, and a second damping leg for the diode.

3.4 Input Stage



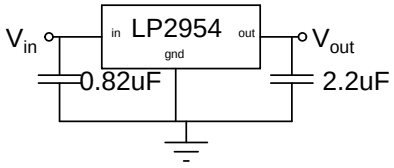
We have a maximum of a $1.5A_{PP}$ square wave on the input at $1MHz$. Our raw input filter had $C = 0.82\mu F$, $L = 40\mu H$, although the value of capacitance was increased by a few μH by bypass capacitors scattered throughout the circuit. This gives a second order roleoff starting at about $52KHz$. At $40dB/decade$, this gives an input ripple of $\approx 1.5A/400$, which is close to three times below spec. In practice, the square wave consists of multiple harmonics, most of which are attenuated far more than the $1MHz$. Bypass caps will also lower this significantly.

To prevent oscillations of the LC filter, combined with the negative impedence input, we put in a damping leg. Assuming the circuit sucks constant power: $V = P/I$ and $R = dV/dI$. This maximize to 12Ω, so we used a 10Ω in the damping leg.

As an "on" indicator, we used a red LED. We added a power switch for convenience in debugging. Strictly, placing the switch after the input filter would give a cleaner power-up transient on both sides, but this set up was more convenient for construction and testing purpose (we added it after we had the input filter built, and we were able to maintain the measure point for V_{in} prior to the input inductor drop).

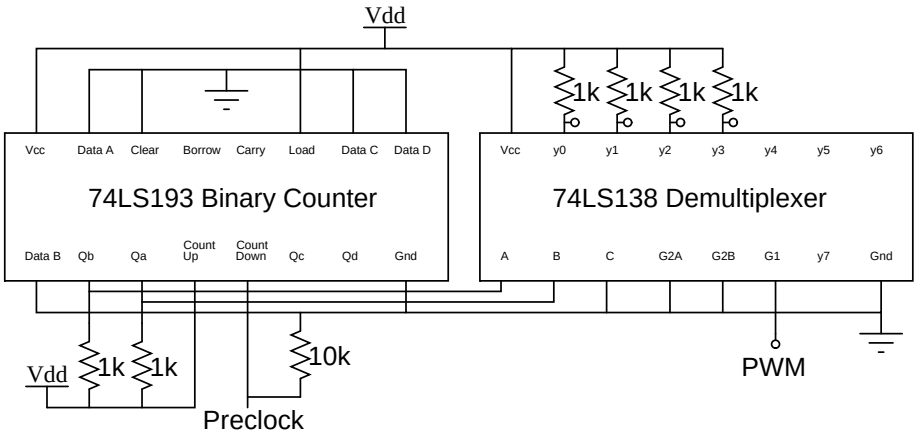
4 Low voltage stages

All of the control systems run off of a regulated 5V. We use the following circuit to regulate:



4.1 Splitter

As mentioned, we have four output switches. Every fourth pulse goes to every fourth switch. To split up the PWM signal, we use the following circuit:



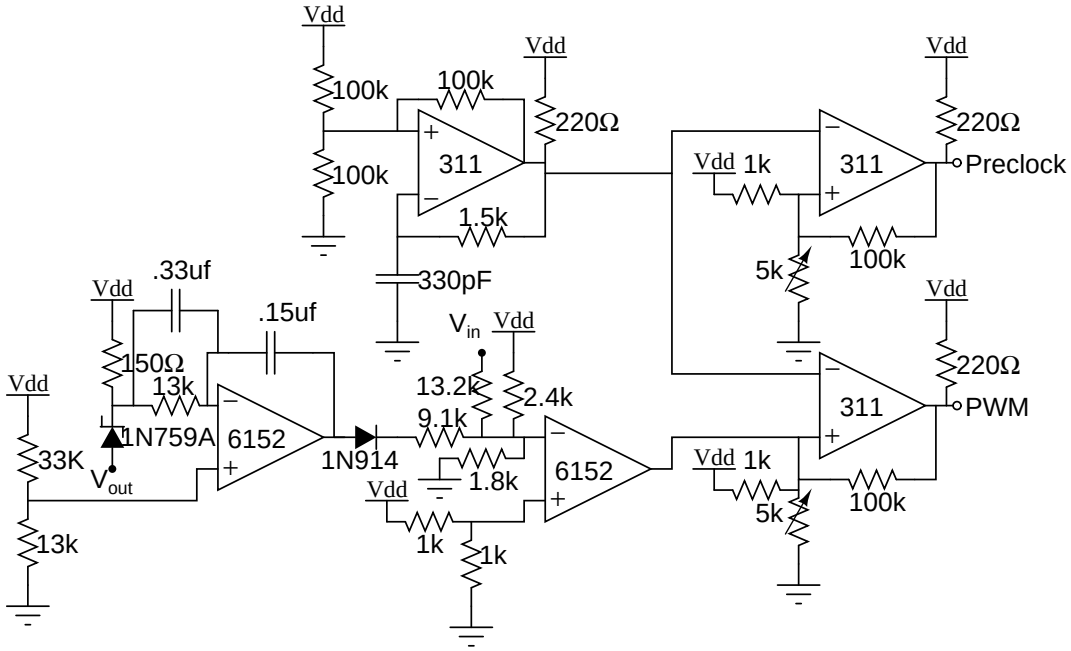
The splitter is just a counter connected to a demultiplexer. The counter selects which switch the input goes to. The demultiplexer actually passes the pulses through to the right switch.

It takes two inputs, a preclock that is used for the digital logic to switch to the right output, and a PWM signal that is then passed to the switch. The preclock must come in between PWM pulses, while all of the switches are turned off.

We use 1k pull-up resistors to speed up the output waveform, as the logic family has much higher sink than source current.

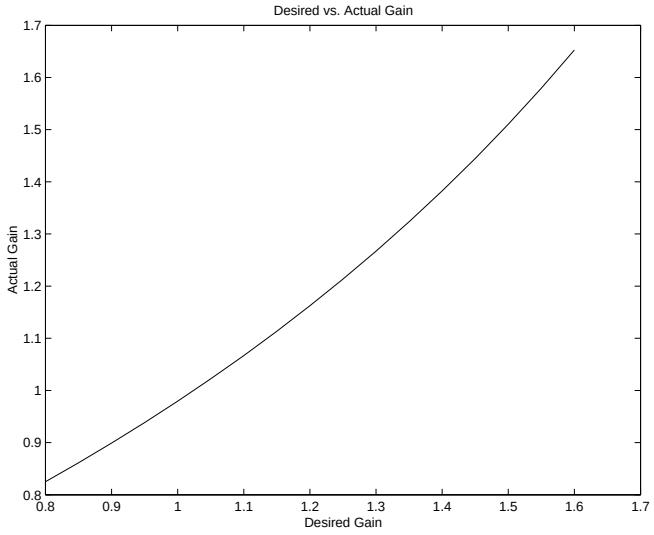
4.2 PWM Topology

Our PWM generator is as follows:



The top left 311 is just a 1MHz Schmidtt trigger oscillator. The bottom left 311 is the compensator. The bottom middle 6152 scales and clips the output of the compensator. The pair of 311s compare the output voltages of the oscillator to the reference voltages to generate the actual PWM and preclock signals.

The clipping is done as follows: the bottom middle 6152 uses a resistor network to V_{in} to set the right PWM frequency open-loop for the input voltage, assuming no transients. We want $D = \frac{V}{1-V}$, which we approximate as $D = a + Vc$. For our input voltage range, this linear approximation is accurate to within about 4%:



In practice, this error is a bit larger, because:

- The oscillator does not put out a perfect triangle wave, and changing voltage changes duty cycle by different amounts at different voltages
- Our resistor values are not perfectly fitted

In practice, we chose our resistor values to give output voltages as just over 10V. We fed the input from the compensator through a diode, so that it could not increase this voltage, but only pull it down to 10V. We also put in a 9.1k resistor, so that it could not pull it too low (in practice, to below about 5V). This was designed so that even if the feedback loop went completely unstable, the output stages would not blow up.

The bottom middle 6152 is strictly redunant, as we could work out the polarities on the resistor networks to accomplish the same task between the compensator and the output comparator. We used it because the 6152 package comes with two op-amps, and using it conceptually simplified the design.

The bottom left operational amplifier is the compensator. We bring the output up by a zener drop, compare it to a reference voltage, and feed in.

4.2.1 Controls

As derived in the initial section,

$$\tilde{v}_{out} = \frac{ILs - D_3 V_{OUT}}{LCs^2 + \frac{Ls}{R} + D_3^2} d_3 + \frac{D_3 V_{IN}}{LCs^2 + \frac{Ls}{R} + D_3^2} d_1$$

For this topology, d_3=$1 - d_1$, so:

$$\tilde{v}_{out} = \frac{ILs - D_3 V_{OUT}}{LCs^2 + \frac{Ls}{R} + D_3^2} d_3 + \frac{D_3 V_{IN}}{LCs^2 + \frac{Ls}{R} + D_3^2} (1 - d_3)$$

This gives two poles and a right half-plane zero. With our values of L and C (in Matlab, computed to be 20% greater than calculated), we found we could compensate this pretty easily with a simple PI compensator. We had a very slow integral term, and a proportional term to give the desired bandwidth.

We got about 20KHz bandwidth. We could not trade off this bandwidth for gain, since any lower proportional gain and we would not have had enough negative feedback at 8KHz. We barely met the overshoot spec of 10 percent at 16V, and then only for very small inputs (the overshoot up changed significantly with step amplitude).